

WEST Search History

DATE: Saturday, March 31, 2007

Hide?	Set Name	Query	Hit Count
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=OR</i>	
<input type="checkbox"/>	L25	(l22 or l23) and ((journal\$ or log\$) with (name or names))	0
<input type="checkbox"/>	L24	(l22 or l23) and l6	0
<input type="checkbox"/>	L23	O'SULLIVAN-BRYAN.in.	2
<input type="checkbox"/>	L22	WALSH-ROBERT.in.	21
<input type="checkbox"/>	L21	l20 and (version near number)	2
<input type="checkbox"/>	L20	l19 and ((journal\$ or log\$) with (name or names))	12
<input type="checkbox"/>	L19	(l17 or l18) and l6	70
<input type="checkbox"/>	L18	707/200-203.ccls.	6182
<input type="checkbox"/>	L17	707/100.ccls.	5184
<input type="checkbox"/>	L16	l14 and (version near number)	11
<input type="checkbox"/>	L15	l14 and (read\$ near (journal\$ or log\$))	0
<input type="checkbox"/>	L14	l6 and ((journal\$ or log\$) with (name or names))	63
<input type="checkbox"/>	L13	l11 and ((journal or log or logs or logging) with (internal or external) with (name or naming or names or label or labels or labeling or tag or tags))	0
<input type="checkbox"/>	L12	l11 and ((journal or log or logs or logging) near (internal or external) near (name or naming or names or label or labels or labeling or tag or tags))	0
<input type="checkbox"/>	L11	l6 and L10	52
<input type="checkbox"/>	L10	((revision or chang\$ or updat\$ or alter\$ or modif\$) near (journal or log or logs or logging))	5748
<input type="checkbox"/>	L9	((revision adj1 control) near (journal or log or logs or logging))	0
<input type="checkbox"/>	L8	((revision adj1 control) with (journal or log or logs or logging))	23
<input type="checkbox"/>	L7	((revision adj1 control) with (journal or log or logs or logging) with ((internal or external) near (name or naming or names or label or labels or labeling or tag or tags)) with version\$)	0
<input type="checkbox"/>	L6	(revision adj1 control)	649
		<i>DB=PGPB,USPT,USOC; PLUR=NO; OP=OR</i>	
<input type="checkbox"/>	L5	L1 and (amend\$ or review\$ or chang\$ or alter\$ or modif\$ or revis\$).ab.	14
<input type="checkbox"/>	L4	L1 and (amend\$ or review\$ or chang\$ or alter\$ or modif\$).ti.	1
<input type="checkbox"/>	L3	L1 and revision.ab.	1
<input type="checkbox"/>	L2	L1 and revision.ti.	1

(5729743 5745906 5787444 5893119 6237041 5265245 5432917 5930798
6021410 4805209 4942602 4943996 4949373 5355493 5375234 5437038

10/ 797,977

5495606 5603025 5640559 5646862 5664186 5668958 5727158 5729744
5752249 5781735 5787416 5805889 5812130 5819271 5826265 5832218
☐ L1 5838918 5842212 5864875 5864871 5878408 5896494 5898874 5899987 50
5909544 5920873 5920867 5923833 5950201 5956710 5958050 5966707
5991897 5991365).pn.

END OF SEARCH HISTORY


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction sur](#)

Terms used

journal and user and internal name and external name and update and version number and revision control

 Sort results by
☒ [Save results to a Binder](#)

 Try an [Advanced Search](#)

 Display results
☒ [Search Tips](#)

 Try this search in [The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐

1 [Fast detection of communication patterns in distributed executions](#)

Thomas Kunz, Michiel F. H. Seuren

 November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research CASCON '97**

Publisher: IBM Press

 Full text available: [pdf\(4.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences non-trivial commun ...

2 [Special issue on persistent object systems: Orthogonally persistent object systems](#)

Malcolm Atkinson, Ronald Morrison

 July 1995 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 4, Issue 3


Publisher: Springer-Verlag New York, Inc.

 Full text available: [pdf\(5.02 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Persistent Application Systems (PASs) are of increasing social and economic importance. They have the potential to be long-lived, concurrently accessed, and consist of large bodies of data and programs. Typical examples of PASs are CAD/CAM systems, office automation, CASE tools, software engineering environments, and patient-care support systems in hospitals. Orthogonally persistent object systems are intended to provide improved support for the design, construction, maintenance, and operation of ...

Keywords: database programming languages, orthogonal persistence, persistent application systems, persistent programming languages

3 [The elements of nature: interactive and realistic techniques](#)

 Oliver Deussen, David S. Ebert, Ron Fedkiw, F. Kenton Musgrave, Przemyslaw Prusinkiewicz, Doug Roble, Jos Stam, Jerry Tessendorf

 August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

10/7997,977

Full text available:  pdf(17.65 MB)

Additional Information: [full citation](#), [abstract](#)

This updated course on simulating natural phenomena will cover the latest research and production techniques for simulating most of the elements of nature. The presenters will provide movie production, interactive simulation, and research perspectives on the difficult task of photorealistic modeling, rendering, and animation of natural phenomena. The course offers a balance of the latest interactive graphics hardware-based simulation techniques and the latest physics-based simulation techni ...

4 Transparent adaptation of single-user applications for multi-user real-time collaboration



Chengzheng Sun, Steven Xia, David Sun, David Chen, Haifeng Shen, Wentong Cai
December 2006 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 13 Issue 4

Publisher: ACM Press

Full text available:  pdf(3.12 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Single-user interactive computer applications are pervasive in our daily lives and work. Leveraging single-user applications for supporting multi-user collaboration has the potential to significantly increase the availability and improve the usability of collaborative applications. In this article, we report an innovative *Transparent Adaptation* (TA) approach and associated supporting techniques that can be used to convert existing and new single-user applications into collaborative ones, ...

Keywords: Application sharing, CoPowerPoint, CoWord, computer-supported cooperative work, operational transformation, transparent adaptation

5 An open-source CVE for programming education: a case study: An open-source CVE for programming education: a case study



Andrew M. Phelps, Christopher A. Egert, Kevin J. Pierre, David M. Parks
July 2005 **ACM SIGGRAPH 2005 Courses SIGGRAPH'05**

Publisher: ACM Press

Full text available:  pdf(7.92 MB)

Additional Information: [full citation](#), [references](#)

6 Special issue on knowledge representation



Ronald J. Brachman, Brian C. Smith
February 1980 **ACM SIGART Bulletin**, Issue 70

Publisher: ACM Press

Full text available:  pdf(13.13 MB)

Additional Information: [full citation](#), [abstract](#), [citations](#)


In the fall of 1978 we decided to produce a special issue of the SIGART Newsletter devoted to a survey of current knowledge representation research. We felt that there were two useful functions such an issue could serve. First, we hoped to elicit a clear picture of how people working in this subdiscipline understand knowledge representation research, to illuminate the issues on which current research is focused, and to catalogue what approaches and techniques are currently being developed. Second ...

7 Final report of the ANSI/X3/SPARC DBS-SG relational database task group



July 1982 **ACM SIGMOD Record**, Volume 12 Issue 4

Publisher: ACM Press


Full text available:  pdf(4.69 MB)


Additional Information: [full citation](#), [citations](#)

8 Human-computer interface development: concepts and systems for its management

H. Rex Hartson, Deborah Hix

March 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 1


 **Publisher:** ACM Press

Full text available:  pdf(7.97 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Human-computer interface management, from a computer science viewpoint, focuses on the process of developing quality human-computer interfaces, including their representation, design implementation, execution, evaluation, and maintenance. This survey presents important concepts of interface management: dialogue independence, structural modeling, representation, interactive tools, rapid prototyping, development methodologies, and control structures. *Dialogue independence* is the ...

9 Revised report on the algorithmic language scheme

 H. Abelson, R. K. Dybvig, C. T. Haynes, G. J. Rozas, N. I. Adams, D. P. Friedman, E. Kohlbecker, G. Steele, D. H. Bartley, R. Halstead, D. Oxley, G. J. Sussman, G. Brooks, C. Hanson, K. M. Pitman, M. Wand

July 1991 **ACM SIGPLAN Lisp Pointers**, Volume IV Issue 3

Publisher: ACM Press

Full text available:  pdf(4.08 MB)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)


The report gives a defining description of the programming language Scheme. Scheme is a statically scoped and properly tail-recursive dialect of the Lisp programming language invented by Guy Lewis Steele Jr. and Gerald Jay Sussman. It was designed to have an exceptionally clear and simple semantics and few different ways to form expressions. A wide variety of programming paradigms, including imperative, functional, and message passing styles, find convenient expression in Scheme.

10 Version models for software configuration management

 Reidar Conradi, Bernhard Westfechtel

June 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 2

Publisher: ACM Press


Full text available:  pdf(483.54 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

After more than 20 years of research and practice in software configuration management (SCM), constructing consistent configurations of versioned software products still remains a challenge. This article focuses on the version models underlying both commercial systems and research prototypes. It provides an overview and classification of different versioning paradigms and definitions and relates fundamental concepts such as revisions, variants, configurations, and changes. In particular, we focus on ...


Keywords: changes, configuration rules, configurations, revisions, variants, versions

11 Types and persistence in database programming languages

 Malcolm P. Atkinson, O. Peter Buneman

June 1987 **ACM Computing Surveys (CSUR)**, Volume 19 Issue 2


Publisher: ACM Press

Full text available:  pdf(7.91 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)


Traditionally, the interface between a programming language and a database has either been through a set of relatively low-level subroutine calls, or it has required some form of embedding one language in another. Recently, the necessity of integrating database and programming language techniques has received some long-overdue recognition. In response, a number of attempts have been made to construct programming languages with completely integrated database management systems. These languages ...

12 Real-time shading

 Marc Olano, Kurt Akeley, John C. Hart, Wolfgang Heidrich, Michael McCool, Jason L. Mitchell, Randi Rost

August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available:  pdf(7.39 MB)

Additional Information: [full citation](#), [abstract](#)

Real-time procedural shading was once seen as a distant dream. When the first version of this course was offered four years ago, real-time shading was possible, but only with one-of-a-kind hardware or by combining the effects of tens to hundreds of rendering passes. Today, almost every new computer comes with graphics hardware capable of interactively executing shaders c thousands to tens of thousands of instructions. This course has been redesigned to address tod real-time shading capabili ...

13 Spoken dialogue technology: enabling the conversational user interface

 Michael F. McTear

March 2002 **ACM Computing Surveys (CSUR)**, Volume 34 Issue 1

Publisher: ACM Press


Full text available:  pdf(987.69 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Spoken dialogue systems allow users to interact with computer-based applications such as databases and expert systems by using natural spoken language. The origins of spoken dialogue systems can be traced back to Artificial Intelligence research in the 1950s concerned with developing conversational interfaces. However, it is only within the last decade or so, with major advances in speech technology, that large-scale working systems have been developed and, in some cases, introduced into commerce ... **NOTES SIGGRAPH '04**


Keywords: Dialogue management, human computer interaction, language generation, language understanding, speech recognition, speech synthesis

14 Data base directions: the next steps

 John L. Berg

November 1976 **ACM SIGMOD Record**, **ACM SIGMIS Database**, Volume 8, 8 Issue 4, 2

Publisher: ACM Press


Full text available:  pdf(9.95 MB)

Additional Information: [full citation](#), [abstract](#), [citations](#)

What information about data base technology does a manager need to make prudent decisions about using this new technology? To provide this information the National Bureau of Standards and the Association for Computing Machinery established a workshop of approximately 80 experts in five major subject areas. The five subject areas were auditing, evolving technology, government regulations, standards, and user experience. Each area prepared a report contained in these proceedings. The proceedings p ...

Keywords: DBMS, auditing, cost/benefit analysis, data base, data base management, government regulation, management objectives, privacy, security, standards, technology assessment, user experience

15 Revised report on the algorithmic language scheme

 J Rees, W Clinger

December 1986 **ACM SIGPLAN Notices**, Volume 21 Issue 12

Publisher: ACM Press

Full text available:  pdf(4.06 MB)

Additional Information: [full citation](#), [citations](#), [index terms](#)

16 IS '97: model curriculum and guidelines for undergraduate degree programs in information systems



Gordon B. Davis, John T. Gorgone, J. Daniel Couger, David L. Feinstein, Herbert E. Longenecker
December 1996 **ACM SIGMIS Database , Guidelines for undergraduate degree programs on Model curriculum and guidelines for undergraduate degree programs in information systems IS '97**, Volume 28 Issue 1

Publisher: ACM Press

Full text available: [pdf\(7.24 MB\)](#)

Additional Information: [full citation](#), [citations](#)

17 Proceedings of the SIGNUM conference on the programming environment for development of numerical software



March 1979 **ACM SIGNUM Newsletter**, Volume 14 Issue 1

Publisher: ACM Press

Full text available: [pdf\(5.02 MB\)](#)

Additional Information: [full citation](#)

18 Interoperability of multiple autonomous databases



Witold Litwin, Leo Mark, Nick Roussopoulos
September 1990 **ACM Computing Surveys (CSUR)**, Volume 22 Issue 3

Publisher: ACM Press

Full text available: [pdf\(2.66 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Database systems were a solution to the problem of shared access to heterogeneous files created by multiple autonomous applications in a centralized environment. To make data usage easier, files were replaced by a globally integrated database. To a large extent, the idea was successful and many databases are now accessible through local and long-haul networks. Unavoidably, users now need shared access to multiple autonomous databases. The question is what the corresponding methodology ...

19 Crowd and group animation



Daniel Thalmann, Christophe Hery, Seth Lippman, Hiromi Ono, Stephen Regelous, Douglas Sutton
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available: [pdf\(20.19 MB\)](#)

Additional Information: [full citation](#), [abstract](#)

A continuous challenge for special effects in movies is the production of realistic virtual crowds, terms of rendering and behavior. This course will present state-of-the-art techniques and methods. The course will explain in details the different approaches to create virtual crowds: particle systems with flocking techniques using attraction and repulsion forces, copy and pasting techniques, agent-based methods. The architecture of software tools will be presented including the MASSIVE software ...

20 Federated database systems for managing distributed, heterogeneous, and autonomous databases



Amit P. Sheth, James A. Larson
September 1990 **ACM Computing Surveys (CSUR)**, Volume 22 Issue 3

Publisher: ACM Press

Full text available: [pdf\(5.02 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A federated database system (FDBS) is a collection of cooperating database systems that are

autonomous and possibly heterogeneous. In this paper, we define a reference architecture for distributed database management systems from system and schema viewpoints and show how various FDBS architectures can be developed. We then define a methodology for developing one of the popular architectures of an FDBS. Finally, we discuss critical issues related to developing an operating an FDBS.

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

ProQuest

[Return to the USPTO NPL Page](#) | [Help](#)

Basic

Advanced

Topics

Publications

My Research
0 marked items

Interface language:

English

Databases selected: Multiple databases...

Results – powered by ProQuest® Smart Search**Suggested Topics** [About](#)< Previous | [Next](#) >[Control](#)[Control AND Organizational behavior](#)[Control AND Models](#)[Control AND Book reviews](#)**Browse Suggested Publications** < Previous | [Next](#) >[Academy of Management. The Academy of](#)[Management Review; Briarcliff Manor](#)[College Composition and Communication; Urbana](#)[Journal of International Business Studies; Washington](#)[Personality and Social Psychology Bulletin; Thousand Oaks](#)4 documents found for: *revision control and journal* » [Refine Search](#) | [Set Up Alert](#)

All sources Scholarly Journals Trade Publications

☐ Mark
all 0 marked items: Email / Cite /
Export Show only full
textSort results by: [Most recent first](#)

- ☐ 1. **Why rein in Linux?**
Robert C Norris Jr. Information Systems Control Journal. Rolling Meadows: 2000. Vol. 4; p. 19
 [Abstract](#)
-
- ☐ 2. **Instituting lease preparation controls**
Miller, Richard A. Journal of Property Management. Chicago: Sep/Oct 1995. Vol. 60, Iss. 5; p. 48 (3 pages)
 [Full text](#) [Full Text - PDF](#) [Abstract](#)
-
- ☐ 3. **A Flexible Framework for Cooperative Distributed Software Development**
Narayanaswamy, K., Goldman, Neil M.. The Journal of Systems and Software. New York: Oct 1991. Vol. 16, Iss. 2; p. 97 (9 pages)
 [Link to full text](#) [Abstract](#)
-
- ☐ 4. **Pumping New Life into an Old Friend - Automated Aperture Card Handling Comes of Age**
Greenspan, Don. IMC Journal. Boulder: Third Quarter 1982. Vol. 18, Iss. 3; p. 31 (3 pages)
 [Abstract](#)

1-4 of 4

Want to be notified of new results for this search? [Set Up Alert](#) Results per page: [30](#) **Basic Search** Tools: [Search Tips](#) [Browse Topics](#) [1 Recent Searches](#)


Search


Clear

Database:



Select multiple databases

10/17/97, 9:27

Date range: 

Limit results to: ☐ Full text documents only 

☐ Scholarly journals, including peer-reviewed  [About](#)

 [More Search Options](#) 

Copyright © 2007 ProQuest Information and Learning Company. All rights reserved. [Terms and Conditions](#)

[Text-only interface](#)



Dialog DataStar

[options](#)[logoff](#)[feedback](#)[help](#)[databases](#)[easy search](#)


Advanced Search:

Inspec - 1898 to date (INZZ)



[limit](#)

Search history:





No.	Database	Search term	Info added since	Results	
CP		[Clipboard]		0	-
1	INZZ	journal OR log\$	unrestricted	8084490	show titles
2	INZZ	revision ADJ control	unrestricted	92	show titles
3	INZZ	1 AND 2	unrestricted	48	show titles
4	INZZ	3 AND version	unrestricted	11	show titles

[hide](#) | [delete all search steps...](#) | [delete individual search steps...](#)Enter your search term(s): [Search tips](#) ☐ Thesaurus mapping whole document Information added since: or: none 
(YYYYMMDD)[search](#)☐ Documents with images

Select special search terms from the following list(s):

- ☒ Publication year 1950-
- ☒ Publication year 1898-1949
- ☒ Inspec thesaurus - browse headings 
- ☒ Inspec thesaurus - enter a term 
- ☒ Classification codes A: Physics, 0-1
- ☒ Classification codes A: Physics, 2-3
- ☒ Classification codes A: Physics, 4-5
- ☒ Classification codes A: Physics, 6
- ☒ Classification codes A: Physics, 7
- ☒ Classification codes A: Physics, 8

10/1999, 997

 FILE-ORGANISATION;  PROGRAMMING-ENVIRONMENTS;  SOFTWARE-TOOLS;  TEXT-EDITING.

Classification codes

C6115 Programming-support*;
C6120 File-organisation;
C6130 Data-handling-techniques.

Keywords

programming-environments; **revision-control**; editor; AVL-trees; AVL-dags; **revision-maintenance**;
command-language.

Treatment codes

P Practical.

Language

English.

Publication type

Journal-paper.

Availability

CCCC: 0164-0925/87/0400-0277\$00.75.

Publication year

1987.

Publication date

19870400.

Edition

1987021.

Copyright statement

Copyright 1987 IEE.

PROCEEDINGS SOFTWARE

(c) 2007 The Institution of Engineering and Technology

Full text available at

USPTO Full Text Retrieval Options



☒ **document 10 of 11** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0002527545 20070101.

Title

RCS-a system for **version control**.

Source

Software - Practice and Experience, {Softw-Pract-Exp-UK}, July 1985, vol. 15, no. 7, p. 637-54, 13
refs, CODEN: SPEXBL, ISSN: 0038-0644, UK.

Author(s)

Tichy-W-F.

Author affiliation

Tichy, W.F., Dept. of Comput. Sci., Purdue Univ., West Lafayette, IN, USA.

Abstract

An important problem in program development and maintenance is **version control**, i.e. the task of keeping a software system consisting of many versions and configurations well organised. The **Revision Control System (RCS)** is a software tool that assists with that task. RCS manages revisions of text documents, in particular source programs, documentation, and test data. It automates the storing, retrieval, **logging** and identification of revisions, and provides selection mechanisms for composing configurations. This paper introduces basic **version control** concepts and discusses the practice of **version control** using RCS. For conserving space, RCS stores deltas, i.e. differences between successive revisions. Several delta storage methods are discussed. Usage statistics show that RCS's delta method is space and time efficient. The paper concludes with a detailed survey of **version control** tools.

Descriptors

SOFTWARE-TOOLS; SYSTEM-DOCUMENTATION.

Classification codes

C6110 Systems-analysis-and-programming*;

C6115 Programming-support.

Keywordsconfiguration-management; RCS; **version-control**; program-development; maintenance; **Revision-Control-System**; software-tool; text-documents; source-programs; documentation; test-data; storing; retrieval; **logging**; identification; selection-mechanisms; delta-storage-methods.**Treatment codes**

P Practical.

Language

English.

Publication type

Journal-paper.

Availability

CCCC: 0038-0644/85/070637-18\$01.80.

Publication year

1985.

Publication date

19850700.

Edition

1985021.

Copyright statement

Copyright 1985 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at **USPTO Full Text Retrieval Options**☒ **document 11 of 11** [Order Document](#)**Inspec - 1898 to date (INZZ)****Accession number & update**

0002371617 20070101.

TitleDIFF: a structured programming editor with **revision control**.**Source**

Transactions of the Information Processing Society of Japan, {Trans-Inf-Process-Soc-Jpn-Japan}, 1984, vol. 25, no. 2, p. 268-76, 5 refs, CODEN: JSGRD5, ISSN: 0387-5806, Japan.

Author(s)

Sakai-S, Ochimizu-K.

Author affiliation

Sakai, S., Ochimizu, K., Graduate School of Electronic Sci. & Technol., Shizuoka Univ., Shizuoka, Japan.

Abstract

One computer program has a number of versions resulting from modifications for its functional extension and other purposes. A means of managing these versions is needed. This paper describes a system for **version** management. The topical areas include: (1) document information management methods for understanding an optional **version**, and (2) methods for representing the reference between versions.

Descriptors

SOFTWARE-TOOLS; STRUCTURED-PROGRAMMING.

Classification codes

C6115 Programming-support*.

Keywords

computer-program-version-management; DIFF; structured-programming- editor; revision-control; document-information-management; reference.

Treatment codes

P Practical.

Language

Japanese.

Publication type

Journal-paper.

Publication year

1984.

Publication date

19840000.

Edition

1985003.

Copyright statement

Copyright 1985 IEE.

(c) 2007 The Institution of Engineering and Technology

save

locally as: PDF document



search strategy:

do not include the search strategy



order

copy to
Clipboard

Top - News & FAQs - Dialog

© 2007 Dialog

Dial g DataStar

options

logoff

feedback

help

databases

search
page

titles

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document



search strategy: do not include the search strategy



order

copy to
Clipboard
☒ Select All

- 1 Towards secure multi-sited transactional revision control systems.
- 2 Towards a new standard for allowing concurrency and ensuring consisten
- 3 Comparison of software architecture reverse engineering methods.
- 4 DocMoto (document management).
- 5 Replicated revision control system.
- 6 Cooperation and collaboration assisted by editors.
- 7 A distributed version control system for wide area networks.
- 8 The case for version control.
- 9 An editor for revision control.
- 10 RCS-a system for version control.
- 11 DIFF: a structured programming editor with revision control.

Full text available at



SCIENCE @ DIRECT

USPTO Full Text Retrieval Options

☒ document 1 of 11 Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0009315874 20070305.

Title

Towards secure multi-sited transactional **revision control** systems.

Source

Computer Standards & Interfaces, {Comput-Stand-Interfaces-Netherlands }, March 2007, vol. 29, no. 3, p. 365-75, 16 refs, CODEN: CSTIEZ, ISSN: 0920-5489.
 Publisher: Elsevier, Netherlands.

Author(s)

Ray-I, Junxing-Zhang.

Author affiliation

Ray, I., Dept. of Comput. Sci., Colorado State Univ., Fort Collins, CO, USA.

Abstract

Version control systems play a very important role in maintaining the **revision** history of software and facilitating software evolution. As the software development process is gradually taking the form of a collaborative effort among several teams hosted over widely dispersed sites, centralized **version control** systems are gradually giving way to multi-sited **version control** systems. Ensuring the integrity and consistency of versioned objects in a environment that supports concurrent access, is a difficult problem. The problem is further aggravated by the need to ensure confidentiality of versioned data as well as non-repudiability of origin. In this paper, we identify the security deficiencies of current

revision control systems and propose a model for secure multi-sited **version control**. Then we develop a transaction management system for **revision control** based on the new secure multi-sited **version control** system model. (All rights reserved Elsevier).

Descriptors

 CONFIGURATION-MANAGEMENT;  SECURITY-OF-DATA;  SOFTWARE-MAINTENANCE.

Classification codes

C6110B Software-engineering-techniques*;
C6130S Data-security.

Keywords

multisited-transactional-revision-control-system; version-control- system; software-evolution; software-development-process; security; transaction-management-system; configuration-management.

Treatment codes

P Practical.

Language

English.

Publication type

Journal-paper.

Availability

SICI: 0920-5489(200703)29:3L:365:TSMS; 1-Q.
Publisher identity number: S0920-5489(06)00074-2.

Digital object identifier

10.1016/j.csi.2006.05.007.

Publication year

2007.

Publication date

20070300.

Edition

2007009.

Copyright statement

Copyright 2007 The Institution of Engineering and Technology.

(c) 2007 The Institution of Engineering and Technology *multisited-transactional-revision-control-system*

Full text available at



SCIENCE @ DIRECT[®]

USPTO Full Text Retrieval Options

☒ **document 2 of 11** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0009315873 20070305.

Title

Towards a new standard for allowing concurrency and ensuring consistency in **revision control** systems.

Source

Computer Standards & Interfaces, {Comput-Stand-Interfaces-Netherlands }, March 2007, vol. 29, no. 3, p. 355-64, 8 refs, CODEN: CSTIEZ, ISSN: 0920-5489.
Publisher: Elsevier, Netherlands.

Author(s)

Ray-I, Junxing-Zhang.

Author affiliation



Ray, I., Dept. of Comput. Sci., Colorado State Univ., Fort Collins, CO, USA.

Abstract

Version control systems play a very important role in maintaining the **revision** history of software and facilitating software evolution. As the software development process is gradually taking the form of

a collaborative effort among several teams hosted over widely dispersed sites, centralized **version control** systems are gradually giving way to multi-sited **version control** systems. Ensuring the integrity and consistency of versioned objects in a environment that supports concurrent access, is a difficult problem. The paradigm of transactions has been successfully used in database systems to ensure integrity of objects. In this paper, we look into the transaction management requirements of **version control** systems and propose a new transaction model of **revision control**. (All rights reserved Elsevier).

Descriptors

 CONCURRENCY-CONTROL;  CONFIGURATION-MANAGEMENT;  SOFTWARE-MAINTENANCE.

Classification codes

C6110B Software-engineering-techniques*.

Keywords

revision-control-systems; software-evolution; software-development- process; transaction-management; configuration-management.

Treatment codes

P Practical.

Language

English.

Publication type

Journal-paper.

Availability

SICI: 0920-5489(200703)29:3L:355:TSAC; 1-5.

Publisher identity number: S0920-5489(06)00073-0.

Digital object identifier

10.1016/j.csi.2006.05.008.

Publication year

2007.

Publication date

20070300.

Edition

2007009.

Copyright statement

Copyright 2007 The Institution of Engineering and Technology.

(c) 2007 The Institution of Engineering and Technology

Full text available at



SCIENCE @ DIRECT®

USPTO Full Text Retrieval Options

☒ **document 3 of 11** [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0009029791 20070101.

Title

Comparison of software architecture reverse engineering methods.

Source

Information and Software Technology, {Inf-Softw-Technol-Netherlands}, July 2006, vol. 48, no. 7, p. 484-97, 43 refs, CODEN: ISOTE7, ISSN: 0950-5849.

Publisher: Elsevier, Netherlands.

Author(s)

Stringfellow-C, Amory-C-D, Potnuri-D, Andrews-A, Georg-M.






Author affiliation

Stringfellow, C., Amory, C.D., Potnuri, D., Dept. of Comput. Sci., Midwestern State Univ., Wichita Falls, TX, USA.

Abstract

Problems related to interactions between components is a sign of problems with the software architecture of the system and are often costly to fix. Thus it is very desirable to identify potential architectural problems and track them across releases to see whether some relationships between components are repeatedly change-prone. This paper shows a study of combining two technologies for software architecture: architecture recovery and change dependency analysis based on **version control** information. More specifically, it describes a reverse engineering method to derive a change architecture from **Revision Control System (RCS)** change history. It compares this method to other reverse engineering methods used to derive software architectures using other types of data. These techniques are illustrated in a case study on a large commercial system consisting of over 800 KLOC of C, C++, and microcode. The results show identifiable problems with a subset of the components and relationships between them, indicating systemic problems with the underlying architecture. (All rights reserved Elsevier).

Descriptors

 CONFIGURATION-MANAGEMENT;  OBJECT-ORIENTED-PROGRAMMING;  REVERSE-ENGINEERING;  SOFTWARE-ARCHITECTURE;  SOFTWARE-MAINTENANCE.

Classification codes

C6110B Software-engineering-techniques*;

C6110J Object-oriented-programming.

Keywords

software-architecture; reverse-engineering-method; software-component-interaction-problem; architecture-recovery; change-dependency-analysis; **version-control-information**; change-architecture; **revision- control-system-change-history**; RCS; software-maintainability.

Treatment codes

P Practical.

Language

English.

Publication type

Journal-paper.

Availability

SICI: 0950-5849(200607)48:7L:484:CSAR; 1-P.

Publisher identity number: S0950-5849(05)00084-4.

Digital object identifier

10.1016/j.infsof.2005.05.007.

Publication year

2006.

Publication date

20060700.

Edition

2006033.

Copyright statement

Copyright 2006 The Institution of Engineering and Technology.

(c) 2007 The Institution of Engineering and Technology

Full text available at **USPTO Full Text Retrieval Options**

☒ **document 4 of 11** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0008752610 20070101.

Title

DocMoto (document management).

Source

Document Manager, {Doc-Manag-UK}, Sept.-Oct. 2005, vol. 13, no. 5, p. 22, 1 refs, CODEN: DOMAFS, ISSN: 1351-3222.

Publisher: Business & Technical Communications, UK.

Abstract

DocMoto is a **version control** tool, enabling simple and effective collaboration for small and medium sized businesses. A WebDAV protocol-based **revision control** server, DocMoto can handle any file type that users might throw at it, from Word documents to spreadsheets and image files. Documents can be checked in and out, and all revisions easily viewed and audited. From this angle, DocMoto could be a way for smaller organisations to begin to address compliance concerns.

Descriptors

 BUYERS-GUIDES;  CONFIGURATION-MANAGEMENT;  DOCUMENT-HANDLING;  SMALL-TO-MEDIUM-ENTERPRISES;  SPREADSHEET-PROGRAMS.

Classification codes

D3045 Records-management-systems-for-business-automation*;

D2010 Business-and-professional-IT-applications;

D5010D Computer-selection-guides-for-office-automation.

Keywords

DocMoto; document-management; **version-control-tool**; small-and-medium-sized-businesses; **WebDAV-protocol-based-revision-control-server**; Word-document; spreadsheet; image-file.

Treatment codes

P Practical;

R Product-review.

Language

English.

Publication type

Journal-paper.

Availability

SICI: 1351-3222(200509/10)13:5L:22:DDM; 1-Q.

Publication year

2005.

Publication date

20050900.

Edition

2006006.

Copyright statement

Copyright 2006 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at **USPTO Full Text Retrieval Options**

☒ **document 5 of 11** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0006941332 20070101.

Title

Replicated **revision control** system.

Source

International **Journal** of Parallel and Distributed Systems & Networks, {Int-J-Parallel-Distrib-Syst-Netw-USA}, 2001, vol. 4, no. 1, p. 8-16, 5 refs, ISSN: 1206-2138.

Publisher: Acta Press, USA.

Author(s)

Mishra-S, Ward-J-S.

Author affiliation





Mishra, S., Dept. of Comput. Sci., Colorado Univ., Boulder, CO, USA.

Abstract

Describes the design and implementation of a replicated **revision control** system (RRCS) that is

suitable for use in a wide-area distributed computing environment in which computing nodes may fail at any time and the communication network may undergo temporary communication failures or communication partitions. This system has been developed by extending an existing **revision control** system (RCS). RRCS provides support to the members of a software development group who are geographically distributed and collaborate by using a wide-area network such as the Internet. It efficiently maintains different versions of the source code files that the group members are developing and makes them available on their local machines. The system ensures **version** availability despite communication or processor failures in the distributed system.

Descriptors

 FAULT-TOLERANT-COMPUTING;  GROUPWARE;  SOFTWARE-ENGINEERING;  WIDE-AREA-NETWORKS.

Classification codes

C5620W Other-computer-networks*;
C5470 Performance-evaluation-and-testing;
C6110B Software-engineering-techniques;
C6130G Groupware;
C6150N Distributed-systems-software.

Keywords

replicated-revision-control-system; wide-area-distributed-computing- environment; computing-nodes; temporary-communication-failures; communication-partitions; software-development-group; source-code-files; local-machines; **version-availability**.

Treatment codes

P Practical.

Language

English.

Publication type

Journal-paper.

Availability

SICI: 1206-2138(2001)4:1L:8:RRCS; 1-#.

Publication year

2001.

Publication date

20010000.

Edition

2001022.

Copyright statement

Copyright 2001 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at **USPTO Full Text Retrieval Options**

☒ **document 6 of 11** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0004346202 20070101.

Title

Cooperation and collaboration assisted by editors.

Source

Wirtschaftsinformatik, {Wirtschaftsinformatik-Germany}, Dec. 1992, vol. 34, no. 6, p. 590-8, 29 refs, CODEN: WIINE9, ISSN: 0937-6429, Germany.

Author(s)

Bonin-H-E-G.

Abstract

Day-to-day administrative work is characterized by entering lots of text using editors. However most of the editors do not provide any support for document composition by two or more persons, who in

practice often have different backgrounds regarding position, competence, function, interests, and experience. It is not sufficient to just maintain the final **version** of the document, it is necessary to track all the contributions. The process of cooperation and collaboration must be recorded in an auditable fashion. CSCW systems should not ignore the widespread use of such editors, instead they should build upon them. The author describes requirements for the production of documents in hierarchically oriented organizations (public administration) and outlines possible improvements to the current practice by using standardized markup (SGML) and conventional **revision control** systems.

Descriptors

DESKTOP-PUBLISHING; GROUPWARE; PAGE-DESCRIPTION-LANGUAGES; PUBLIC-ADMINISTRATION; TEXT-EDITING.

Classification codes

C7130 Public-administration*;
C7108 Desktop-publishing;
C5620 Computer-networks-and-techniques;
C6130D Document-processing-techniques.

Keywords

administrative-work; document-composition; cooperation; collaboration; hierarchically-oriented-organizations; public-administration; standardized-markup; SGML; **conventional-revision-control-systems**.

Treatment codes

P Practical.

Language

German.

Publication type

Journal-paper.

Publication year

1992.

Publication date

19921200.

Edition

1993006.

Copyright statement

Copyright 1993 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at [custom link](#) **USPTO Full Text Retrieval Options**

☒ **document 7 of 11** [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0003794374 20070101.

Title

A distributed **version control** system for wide area networks.

Source

Software Engineering **Journal**, {Softw-Eng-J-UK}, Sept. 1990, vol. 5, no. 5, p. 255-62, 20 refs, CODEN: SEJOED, ISSN: 0268-6961, UK.

Author(s)

O'Donovan-B, Grimson-J-B.

Author affiliation

O'Donovan, B., Grimson, J.B., Dept. of Comput. Sci., Trinity Coll., Dublin, Ireland.

Abstract

A distributed **revision control** system (DRCS) that is suitable for use in wide area networks, is described. A selective amount of replication is used to improve performance. The system was developed as an extension to an existing **revision control** system (RCS). DRCS runs on various

versions of the Unix system. It uses the UUCP communication protocol, but it can be easily adapted to use another communications protocol. The system has been used as a tool to **control** the source files for a document that is being jointly authored by two persons who are geographically separated by over 200 km. The performance of the system has been closely monitored, and the results of this monitoring will be used to provide ideas for improvements which will be incorporated into **version 2** of the system.

Descriptors

 COMPUTER-NETWORKS;  PROJECT-SUPPORT-ENVIRONMENTS;  SOFTWARE-ENGINEERING;
 UNIX.

Classification codes

B6210L Computer-communications*;
C6115 Programming-support*;
C6110B Software-engineering-techniques;
C5620W Other-computer-networks.

Keywords

distributed-version-control-system; distributed-revision-control- system; DRCS; wide-area-networks; replication; RCS; Unix-system; UUCP-communication-protocol; source-files.

Treatment codes

P Practical.

Language

English.

Publication type

Journal-paper.

Publication year

1990.

Publication date

19900900.

Edition

1991003.

Copyright statement

Copyright 1991 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at **USPTO Full Text Retrieval Options**

☒ **document 8 of 11** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0003403244 20070101.

Title

The case for **version control**.

Source

EXE, {EXE-UK}, April 1989, vol. 3, no. 10, p. 36-8, 40, 0 refs, CODEN: EXEE5, ISSN: 0268-6872, UK.

Author(s)

Middleditch-M.

Author affiliation




Middleditch, M., Semantics Ltd., Harlow, UK.

Abstract

The central functions of the IPSE, namely the configuration management and **version control** functions, are available as stand alone packages costing a fraction of the price of the cheapest IPSE package. These systems provide a function known variously as configuration management, **version control**, **revision control** and **version** management. A **version control** system (VCS) can have a significant impact in all areas of the software development process-project **control**, quality assurance, development time scales, cost and maintenance. To gain the full benefits, however, it may be

necessary to formalise working practices in the development environment. The author looks at the features of **version control** systems.

Descriptors

 PROJECT-SUPPORT-ENVIRONMENTS;  SOFTWARE-TOOLS;  STORAGE-MANAGEMENT;
 UTILITY-PROGRAMS.

Classification codes

C6115 Programming-support*;
C6120 File-organisation.

Keywords

software-tools; storage-management; utility-programs; project-support-environments; configuration-management; **version-control**; **revision-control**; **version-management**; software-development-process; development-environment.

Treatment codes

P Practical.

Language

English.

Publication type

Journal-paper.

Publication year

1989.

Publication date

19890400.

Edition

1989015.

Copyright statement

Copyright 1989 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at



USPTO Full Text Retrieval Options

☒ **document 9 of 11** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0002979058 20070101.

Title

An editor for **revision control**.

Source

ACM Transactions on Programming Languages and Systems, {ACM-Trans-Program-Lang-Syst-USA}, April 1987, vol. 9, no. 2, p. 277-95, 13 refs, CODEN: ATPSDT, ISSN: 0164-0925, USA.

Author(s)

Fraser-C-W, Myers-E-W.

Author affiliation

Fraser, C.W., Dept. of Comput. Sci., Arizona Univ., Tucson, AZ, USA.

Abstract

Programming environments support **revision control** in several guises. Explicitly, **revision control** software manages the trees of revisions that grow as software is modified. Implicitly, editors retain past versions by automatically saving backup copies and by allowing users to undo commands. This paper describes an editor that offers a uniform solution to these problems by never destroying the old **version** of the file being edited. It represents files using a generalization of AVL trees called AVL dags, which makes it affordable to automatically retain past versions of files. Automatic retention makes **revision** maintenance transparent to users. The editor also uses the same command language to edit both text and **revision** trees.

Descriptors



[Web](#) [Images](#) [Video](#) [News](#) [Maps](#) [more »](#)

journal with revision control with update with in

[Advanced Scholar Search](#)
[Scholar Preferences](#)
[Scholar Help](#)

The following words are very common and were not included in your search: **with with with with with.** [\[details\]](#)

Scholar [All articles](#) [Recent articles](#) Results 1 - 10 of about 20,300 for **journal with revision control with u**

All Results

[G Ganger](#)

[J Homere](#)

[W Gropp](#)

[B Worthington](#)

[Y Patt](#)

Using versioning to support collaboration on the WWW - group of 3 »

F Vitali, DG Durand - World Wide Web **Journal**, 1995 - cs.unibo.it

... published on the World Wide Web **Journal**, O'Reilly ... can be requested to modify and **update** according to ... G. Durand, "Palimpsest, a Data Model for **Revision Control**". ...

Cited by 25 - [Related Articles](#) - [Cached](#) - [Web Search](#)

Method and apparatus for reconciling different versions of a file - group of 3 »

JH Howard - US Patent 5,600,834, 1997 - Google Patents

... may be invoked under user or application **control**, either at ... conflict since neither Version #3 nor Version #4 was ... to the file **name**, the **journal** entries indicate ...

Cited by 73 - [Related Articles](#) - [Web Search](#)

Towards a Uniform Version Model for Software Configuration Management - group of 5 »

R Conradi, B Westfechtel - Software Configuration Management, Proceedings of the ICSE ... - idi.ntnu.no

... The functionality of version **control** is heavily influenced by the way V ... bound v d a evd to an **internal** and fully ... **Revision** chains can be built from constraints of ...

Cited by 28 - [Related Articles](#) - [View as HTML](#) - [Web Search](#) - [BL Direct](#)

Formalizing dynamic software updating - group of 12 »

G Bierman, M Hicks, P Sewell, G Stoye - Proc. 2nd International Workshop on Unanticipated Software ..., 2003 - cs.umd.edu

... updates. Of equal importance is the need to **control** an **update's** effect.

Which mod- ules will 'notice' the new version? Can ...

Cited by 31 - [Related Articles](#) - [View as HTML](#) - [Web Search](#)

Cluster command and control (c3) tool suite - group of 5 »

M Brim, R Flanery, BL Al Geist, S Scott - Proceedings of 3rd Austrian-Hungarian Workshop on ..., 2000 - forge-fre.ornl.gov

... a cluster node and perhaps taking **control** of the ... restricts the usefulness of the image **update** to preplanned ... orca.st.usm.edu/pdcp/) Initial version published in ...

Cited by 14 - [Related Articles](#) - [View as HTML](#) - [Web Search](#)

DNS Overview with a discussion of DNS Spoofing - group of 2 »

S Hanley - ogobin.de

... uses both the TCP (Transmission **Control** Protocol) and ... "Securing DNS and BIND", Linux

Journal, October 2000. ... with FinePrint pdfFactory trial version <http://www...>

Cited by 2 - [Related Articles](#) - [View as HTML](#) - [Web Search](#)

VTML for Fine-Grained Change Tracking in Editing Structured Documents - group of 5 »

L Bendix, F Vitali - Proceedings of the Software Configuration Management ... - Springer

18/797,977

... have looked into traditional tools for version **control**, like RCS ... **name** for CoEd to **update** its **internal** ... The Get_version operation creates the required version. ...
[Cited by 7](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard - group of 16 »

W Gropp, EL Lusk, N Doss, A Skjellum - Parallel Computing, 1996 - [www-unix.mcs.anl.gov](#)
... efficient to use the blocking version for implementing ... to perform the necessary buffer management and flow **control**. ... do maintain their own **internal** buffers and ...
[Cited by 1003](#) - [Related Articles](#) - [View as HTML](#) - [Web Search](#) - [BL Direct](#)

Ninja: A framework for network services - group of 10 »

JR von Behren, EA Brewer, N Borisov, M Chen, M ... - Proceedings of the 2002 USENIX Annual Technical Conference, 2002 - [usenix.org](#)
... way to revert to the previous version easily ... may use partitions for fine-grain **control** of both ... Robust Internet-Scale Systems and Services." **Journal of Computer** ...
[Cited by 27](#) - [Related Articles](#) - [Web Search](#)

Development of the Logical Observation Identifier Names and Codes (LOINC) Vocabulary - group of 6 »

SM Huff, RA Rocha, CJ McDonald, GJE De Moor, T ... - **Journal of the American Medical Informatics Association**, 1998 - 171.66.121.52
... **Journal of the American Medical Informatics Association** 5:276-292 ... There should be version **control** associated with the coding ... The ASN.1 version of the LOINC SDM ...
[Cited by 58](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

Goooooooooooooogle ►

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

journal with revision control with upd **Search**

[Google Home](#) - [About Google](#) - [About Google Scholar](#)

©2007 Google